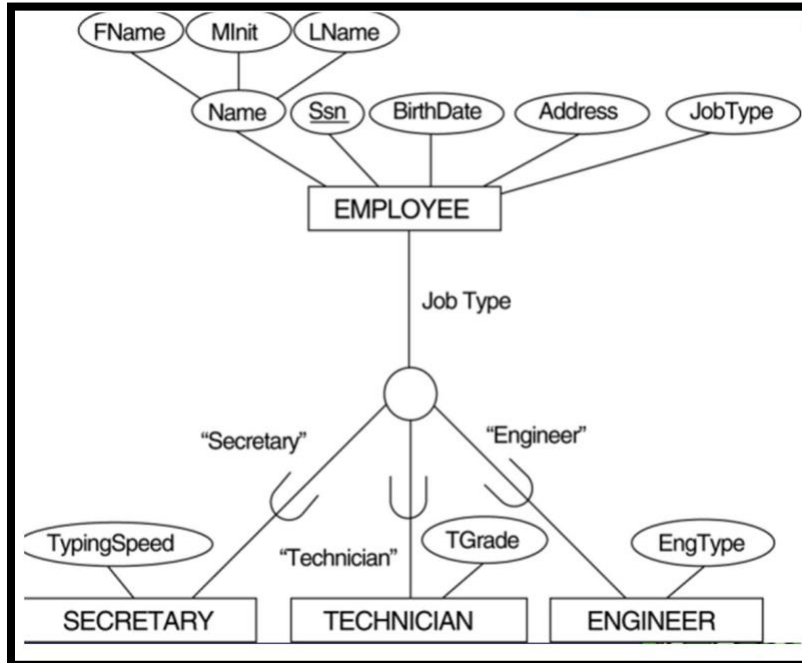


# EER Mapping Exercises

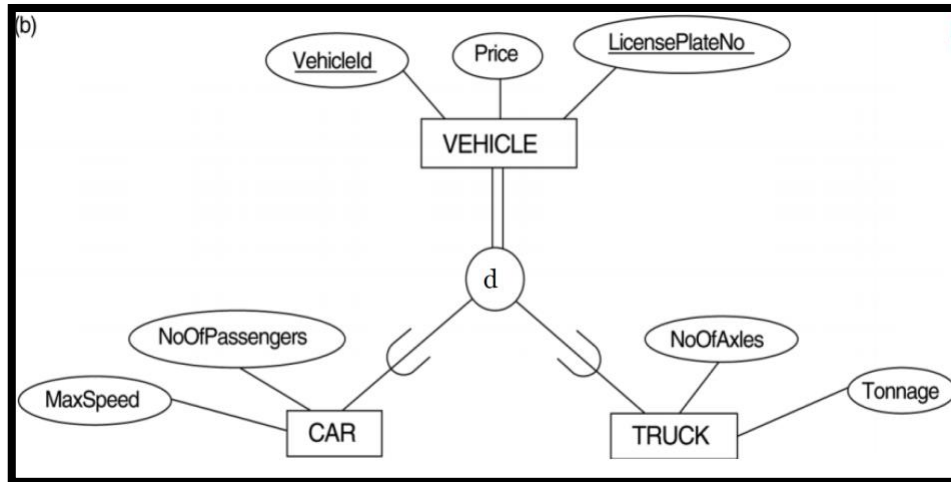
## Option 7A Multiple Relations - Superclass and Subclasses:

Create a table  $L$  for  $C$  with attributes  $(L) = \{k, a_1, \dots, a_n\}$  and  $PK(L) = k$ . Create a relation  $L_i$  for each subclass  $S_i$ ,  $1 \leq i \leq m$ , with the attributes  $(L_i) = \{k\} \cup \{\text{attributes of } S_i\}$  and  $PK(L_i) = k$ . This option works for any specialization (total or partial, disjoint or overlapping).



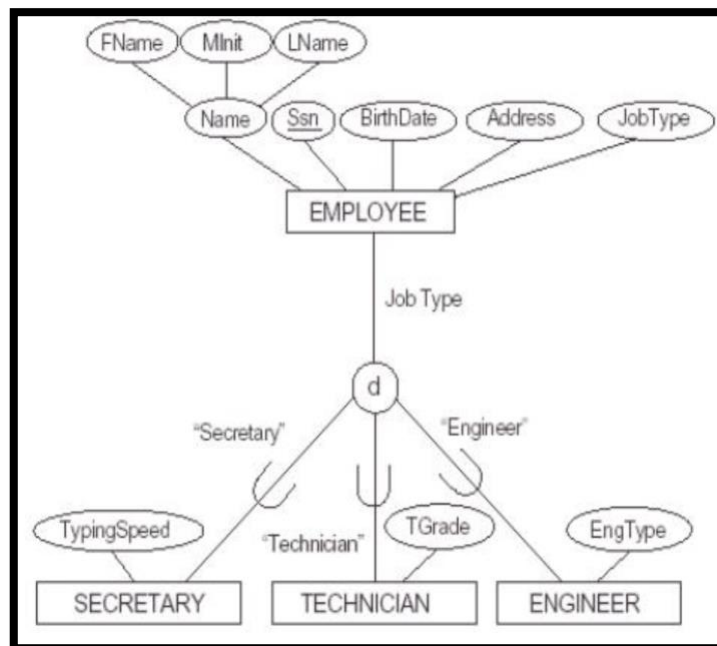
## Option 7B Multiple Relations - Subclass Relation Only:

Create a table  $L_i$  for each subclass  $S_i$ ,  $1 \leq i \leq m$  with the Attributes  $(L_i) = \{\text{attributes of } S_i\} \cup \{k, a_1, \dots, a_n\}$  and  $PK(L_i) = k$ . This option only works for a specialization whose subclasses are total (**Why?**). If the specialization is overlapping; an entity may be duplicated in several relations. (**If the specialization is disjoint & total it will be optimal mapping**).



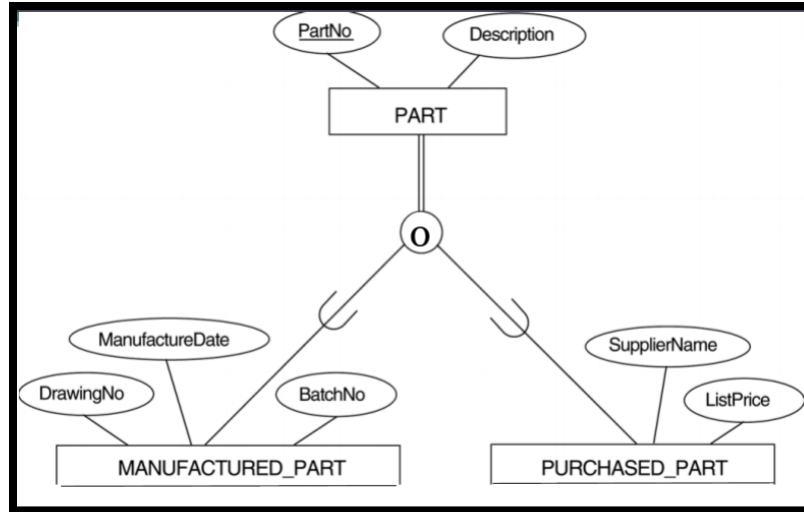
**Option 7C Single Relation with One Type Attribute:**

Create a single table L with attributes  $(L) = \{k, a_1, \dots, a_n\} \cup \{\text{attributes of } S_1\} \cup \dots \cup \{\text{attributes of } S_m\} \cup \{t\}$  and  $PK(L)=k$ . The attribute t is called a type (or discriminating) attribute that indicates the subclass to which each tuple belongs, if any. This Option works only for a specialization whose subclasses are **disjoint** and has the potential for generating many **Null** values if many specific attributes exist in a subclass.



**Option 7D: Single Relation with Multiple Type Attributes:**

Create a single table schema L with Attributes  $(L) = \{k, a_1, \dots, a_n\} \cup \{\text{attributes of } S_1\} \cup \dots \cup \{\text{attributes of } S_m\} \cup \{t_1, t_2, \dots, t_m\}$  and  $PK(L) = k$ .



**Mapping of If all the superclasses have the same key**

Include the key as an attribute of the category.

Otherwise:

1. Create a new key attribute, called a *surrogate key*, as primary key of the category.
2. Add surrogate key as foreign key for each superclass relation of the category
3. Add an attribute type to the category identifying particular entity type of the superclasses (PERSON, BANK, COMPANY) Union Types (Categories)

